Pertemuan 3

Struktur Perulangan pada Java

Objektif :

1. Mahasiswa dapat memahami konsep struktur kontrol perulangan dalam pemrograman.
2. Mahasiswa dapat menggunakan struktur kontrol perulangan (while, do-while, for) yang digunakan.
3. Mahasiswa dapat membuat program yang berisi alur program perulangan melalui contoh kasus.

P3.1 Teori

1. Perulangan

Struktur kontrol pengulangan adalah berupa pernyataan dari Java yang mengijinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah tertentu yang
diinginkan. Ada tiga macam jenis dari struktur kontrol pengulangan yaitu :

1. **While**
2. **Do-while**
3. **For loops**

Pernyataan-pernyataan di atas menciptakan *loop*. *Loop* secara berulang mengeksekusi sebarisan instruksi yang sama sampai kondisi akhir ditemui. Dengan kata lain, *looping* atau *loop* artinya mengulangi eksekusi blok program tertentu sampai tercapai kondisi untuk menghentikannya (terminasi). Setiap perulangan memiliki 4 bagian yaitu :

- ✓ inisialisasi (*initialization*),
- ✓ badan program (*body*) / *statement*,
- ✓ iterasi (*iteration*), dan
- ✓ *termination*.

1.1 Statement While

Pernyataan *while loop* adalah pernyataan atau blok pernyataan yang diulang-ulang sampai mencapai kondisi yang cocok.

Bentuk pernyataan while,

```
while( boolean_expression ){
statement1;
statement2;
. . .
}
```

Pernyataan di dalam *while loop* akan dieksekusi berulang-ulang selama kondisi *boolean_expression* bernilai benar (*true*). Contoh pada kode di bawah ini:

```
int i = 4;
while ( i > 0 ){
System.out.print(i);
```

```
i--; }
```

Contoh di atas akan mencetak angka 4321 pada layar. Perlu dicatat jika bagian i--; dihilangkan, akan menghasilkan pengulangan yang terus menerus (**infinite loop**). Sehingga, ketika menggunakan *while loop* atau bentuk pengulangan yang lain, pastikan Anda memberikan pernyataan yang membuat pengulangan berhenti pada suatu kondisi.

1.2 Statement Do While

Do-while loop mirip dengan *while-loop*. Pernyataan di dalam *do-while loop* akan dieksekusi beberapa kali selama kondisi bernilai benar(*true*). Perbedaan antara *while* dan *do-while loop* adalah dimana pernyataan di dalam *do-while loop* akan dieksekusi sedikitnya **satu kali.**

Bentuk pernyataan do-while,

```
do{
statement1;
statement2;
. . .
}while( boolean_expression );
```

Pernyataan di dalam *do-while loop* akan dieksekusi pertama kali, dan akan dievaluasi kondisi dari *boolean_expression*. Jika nilai pada *boolean_expression* tersebut bernilai *true*, pernyataan di dalam *do-while loop* akan dieksekusi lagi.

Berikut ini beberapa contoh do-while loop:

Contoh 1:

```
int x = 0;
do
{
System.out.println(x);
x++;
}while (x<10);
```

Contoh ini akan memberikan output 0123456789 pada layar.

Contoh 2:

```
//infinite loop
do{
```

```
System.out.println("hello");
} while (true);
```

Contoh di atas akan melakukan pengulangan terus menerus yang menulis kata "hello" pada layar.

Contoh 3:

```
//one loop
// statement is executed once
do
System.out.println("hello");
while (false);
```

Contoh di atas akan memberikan output hello pada layar.

1.3 Statement Perulangan For

Perulangan *for* menyediakan sarana mengulang kode dalam jumlah yang tertentu. Pengulangan ini terstruktur untuk mengulangi kode sampai tercapai batas tertentu.

Berikut bentuk dasar perulangan *for* :

for(InitializationExpression; LoopCondition; StepExpression)

statement

- ✓ **InitializationExpression**, digunakan untuk inisialisasi variabel kendali perulangan.
- ✓ **LoopCondition**, membandingkan variabel kendali perulangan dengan suatu nilai batas.
- ✓ **StepExpression**, menspesifkasikan cara variabel kendali dimodifikasi sebelum iterasi berikutnya dari perulangan.

Contoh:

```
public class For1
{
public static void main(String[] args)
{
int i;
for (i = 1;i<11;i++)
System.out.println(i);
```

```
}
}
```

Hasil (*output*) dari contoh listing program di atas:

1
2
3
4
5
6
7
8
9
10

<u>Pernyataan *for-loop* Bersarang</u>

Java memungkinkan *loop* yang disarangkan di *loop* yang lain. Satu *loop* berada di dalam *loop* yang lainnya.

Contoh:

```
public class Loopbersarang1{
public static void main(String[] args){
for(int i=0;i<10;i++){
for(int j=i;j<10;j++){
System.out.print("*");}
System.out.println();
}
}
}
```

Hasil (*output*) dari contoh listing program di atas:

```
*
**
*
**
*
**
*
```

P3.2 Contoh Kasus

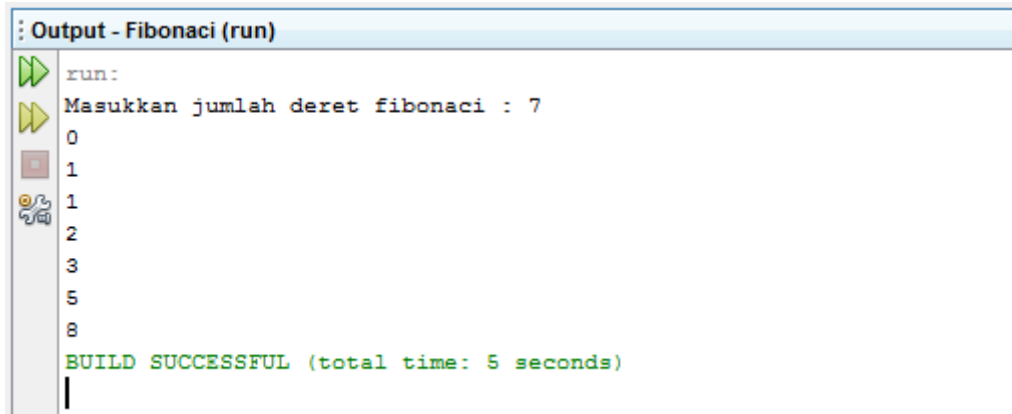
Buat program mencetak segitiga siku-siku menggunakan perulangan for!

Output yang dihasilkan:

Langkah-langkah pengerjaan:

1. Jalankan Netbeans Anda.
2. Lalu buat file project baru dengan memilih menu File – New Project, atau dengan menggunakan hotkey Ctrl+Shift+N.
3. Pilih jenis project yang akan dibuat (Java – Java Application)
4. Tentukan nama project dan lokasi penyimpanan project. Contoh: nama project: SegitigaSiku, dan nama kelas: SegitigaSiku

5. Ketikkan kode program di bawah ini pada code editor

```
package segitigasiku;

import java.io.*;
public class SegitigaSiku {

    public static void main(String[] args) throws Exception{
        DataInputStream masukan = new DataInputStream(System.in);

        String strtinggi = null;

        System.out.println("Segitiga SIku-Siku");
        System.out.print("Masukkan Tinggi : ");
        strtinggi = masukan.readLine();

        int tinggi, i, j, k;
        tinggi = Integer.parseInt(strtinggi);   // mengubah inputan variabel strtinggi dengan
                                                // tipe String ke variabel tinggi dengan tipe data
                                         integer

        /* perulangan for i melakukan perulangan dari 1 sampai sebanyak nilai variabel tinggi
yang
              dimasukkan(perulangan baris / banyaknya baris yang akan dicetak) */
        for(i=1;i<=tinggi;i++){

/* perulangan for j melakukan perulangan spasi(" ")sebanyak nilai variabel tinggi dan
berkurang
1 setiap perulangannya(berpindah baris) sampai jumlah spasi(" ") yang dicetak pada baris
yang
bersangkutan berjumlah 1 */
            for(j=tinggi;j>=i;j--){
                System.out.print(" ");
            }

/* perulangan for k melakukan perulangan bintang("*")dari 1 sampai sebanyak nilai variabel
tinggi dan bertambah 1 setiap perulangannya sampai jumlah bintang("*") yang dicetak sama
dengan nilai variabel tinggi yang diinput  */
            for(k=1;k<=i;k++){
                System.out.print("*");
            }

// perintah berganti baris ini dilakukan setiap selesai 1x dalam melakukan perulangan j dan
perulangan k
            System.out.print("\n");
        }
    }
}
```

6. Build project tersebut dengan memilih menu Run => Build Main Project, atau dengan menggunakan hotkey F11.

7. Jika tidak ada kesalahan (BUILD SUCCESSFUL), jalankan project tersebut dengan memilh menu Run => Run Main Project, atau dengan menggunakan hotkey F6.

P3.3 Latihan

Buat program deret fibonaci. Output yang dihasilkan adalah sebagai berikut:

```
Output - Fibonaci (run)
run:
Masukkan jumlah deret fibonaci : 7
0
1
1
2
3
5
8
BUILD SUCCESSFUL (total time: 5 seconds)
```

Jawaban:

1. Jalankan Netbeans Anda
2. Lakukan langkah-langkah pengerjaan seperti contoh kasus sebelumnya.
3. Pada code editor Netbeans, ketikkan program berikut:

```
cpackage fibonaci;

/**
 *
 * @author Shinigami
 */
import java.io.*;
public class Fibonaci {

    public static void main(String[] args) throws Exception{
        BufferedReader masuk = new BufferedReader(new InputStreamReader(System.in));

        String jumlahfibo = null;
        int a=0, b=1, c=0, i=1;

            System.out.print("Masukkan jumlah deret fibonaci : ");
            jumlahfibo = masuk.readLine();

            int jumlah;
            jumlah = Integer.parseInt(jumlahfibo);

                    System.out.println(a);
                    System.out.println(b);
                    do {
                            c = a + b;
                            System.out.println(c);
                            a = b;
                            b = c;
                i++;
                    } while (i < (jumlah-1));
            }
    }
```

P3.4 Daftar Pustaka

Naughton, Patrick, *Java Handbook: Konsep Dasar Pemrograman Java*, Andi Yogyakarta, 1996.

Gary Cornell dan Cay S.Horstmann, *Core Java edisi Indonesia*, Andi, Yogyakarta, 1997.

ANuff, *Penuntun Pemrograman Java*, Andi Yogyakarta, 1997.

Abdul Kadir, *Dasar Pemrograman Java 2*, Andi Yogyakarta, 2008.